

TP Informatique 4

Tous les exercices seront traités dans la même feuille de script que l'on nommera par exemple `TP04.py`, à enregistrer dans votre espace personnel (disque U:). Ce script sera exécuté (touche F5) pour que les fonctions saisies dans le script soient utilisables dans la console.

Exercice 1

Dans l'éditeur, saisir :

```
a,b=1,2

def swap(a,b):
    a,b=b,a
    return a,b

print("a=",a,"b=",b)
print("swap(a,b)=",swap(a,b))
print("a=",a,"b=",b)
```

1. Prédire puis vérifier les affichages réalisés par le script proposé.
2. Critiquer le procédé utilisé dans la fonction `swap`.

Exercice 2

Dans l'éditeur, saisir :

```
g=9.81
import numpy as np
```

En négligeant les frottements de l'air, écrire une fonction `vit(h)` d'argument la hauteur `h` en mètres et qui renvoie sa vitesse $\sqrt{2gh}$ lors de l'impact au sol en m/s .

Exercice 3

Dans l'éditeur, saisir :

```
def impair(n):
    print(locals())
    r=n%2
    return r==1
```

1. Tester la fonction sur différents entiers. Quel est le statut de l'argument `n` pour la fonction `impair` ?
2. Déplacer l'instruction `print(locals())` juste avant le `return` puis tester la fonction sur différents entiers. Qu'observe-t-on ?

Exercice 4

1. Dans l'éditeur, écrire une fonction `taille(n)` d'argument `n` un entier et qui renvoie le nombre de chiffres de l'écriture décimale de n .
2. Tester cette fonction et vérifier notamment que 2^{2025} est bien de taille 610.

Exercice 5

1. Écrire une fonction `eq(f,a,b)` d'arguments `a` et `b` des flottants distincts, `f` une fonction définie sur un intervalle I et qui renvoie les coefficients m et p de la droite passant par les points $(a, f(a))$ et $(b, f(b))$ d'équation $y = mx + p$.
2. Tester la fonction `eq` sur la fonction $f(x) = x^2$ et $a = 1$, $b = 2$ et éventuellement d'autres fonctions f .

Exercice 6

1. Écrire une fonction `lyc(n)` d'argument un entier `n` qui renvoie `True` si l'entier `n` additionné à l'entier dont l'écriture décimale est celle de `n` renversé est un nombre palindrome et `False` sinon. Par exemple, avec le nombre 19, on réalise les opérations :

$$19 \rightarrow 19 + 91 = 110 \neq 011 \rightarrow \text{False}$$

Avec le nombre 145, on réalise les opérations

$$145 \rightarrow 145 + 541 = 686 \rightarrow \text{True}$$

On définira une fonction locale `sym(p)` dans la fonction `lyc` d'argument `p` entier et qui renvoie l'entier dont l'écriture décimale est celle de `p` renversée.

2. Modifier le code précédent pour afficher les variables locales de `lyc` puis de `sym`. Tester la fonction `lyc` sur différents entiers.

Exercice 7

Écrire une fonction `numjour(date)` d'argument une chaîne `date` au format "JJ/MM" qui renvoie `True` si la date est cohérente et `False` sinon. On supposera que la saisie du mois est valide et que l'année n'est pas bissextile.

Exercice 8

Dans l'éditeur, écrire une fonction `uni(L)` d'argument `L` une liste qui renvoie `True` si la liste contient deux éléments distincts et `False` sinon. Tester la fonction sur différentes listes.

Exercice 9

1. Dans l'éditeur, écrire une fonction `renv1(L)` d'argument `L` une liste qui renvoie la liste `L` renversée. La fonction doit être codée sans effet de bord et renvoie donc une nouvelle liste.
2. Dans l'éditeur, écrire une fonction `renv2(L)` d'argument `L` une liste, qui agit *en place* sur l'objet fourni en argument et le renverse directement. La fonction est avec effet de bord, elle ne renvoie rien.

On testera les deux fonctions sur différentes listes.